REMARKS

Claims 1, 4-12, 31-39, 42-44, 46-51, 54-56, and 58-62 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 103(a) Rejection:

The Examiner rejected claims 1, 4-12, 31-39, 42-44, 46-51, 54-56 and 58-62 under 35 U.S.C. § 103(a) as being unpatentable over "The Error Handling Interface (H5E)" (hereinafter "H5E-A") in view of "H5E: Error Interface" (hereinafter "H5E-B"). Applicants respectfully traverse this rejection for at least the following reasons.

In regard to claim 1, the cited art does not teach or suggest *in each of two or more threads of a multithreaded program, for each error generated by one or more functions executed in the thread, store an error trace element in a memory storage area private to the thread in accordance with an application programming interface (API) to an error trace mechanism.* As illustrated in Fig. 2 and described in the accompanying description (e.g., paragraph [0021]) of Applicant's specification, each of the two or more threads in the multithreaded program is associated with a particular, separate and distinct, "thread private data", or "private data area", for the thread, which is exemplary of a memory storage area private to the corresponding thread. Error trace elements generated for a particular thread are stored to the corresponding private data area, a memory storage area private to the thread, as is clearly illustrated in Fig. 2 and disclosed in paragraph [0021]:

> Each thread private data 100 is a storage area (memory) specific to a particular thread. Each thread may store and access data in its associated thread private data 100 for the thread. In one embodiment, when error traces are stored using thread private data, all error trace elements generated for a particular thread are recorded in the thread's private data area. Thus, each thread private data 100 may store error traces 104, if any, for its associated thread.

The H5E-A reference does not teach or suggest an error trace mechanism for a multithreaded program configured to, for each error generated by one or more functions executed in each of **two or more threads** in the multithreaded program, store an error trace element in a memory storage area **private to** the corresponding thread, as is recited in claim 1. Further, the H5E-A reference does not teach or suggest an error trace mechanism configured to obtain an error trace for each of **two or more threads** of a multithreaded program, wherein each error trace includes one or more error trace elements specific to the corresponding thread, as is recited in claim 1.

Examiner asserts H5E-B discloses "two or more threads of a multithreaded program and storing an error trace element in a memory area private to the thread for each of the two or more threads." Examiner cites a portion of page 1, paragraph 7, as "Each thread has its own error stack…multi-threading…" **However, Examiner failed to cite the entire paragraph, which actually states** (emphasis added):

> Each thread has its own error stack, but since multi-threading has not been added to the library yet, this package maintains a single error stack. The error stack is statically allocated to reduce the complexity of handling errors within the H5E package.

From the above, while H5E-B does disclose that each thread has its own error stack, contrary to the Examiner's assertion, H5E-B does not teach "storing an error trace element in a memory area private to the thread for each of the two or more threads." H5E-B actually teaches that the "package" maintains a single error stack. Furthermore, contrary to the Examiner's assertion, H5E-B does not teach support for multi-threaded programs. H5E-B actually teaches that "multi-threading has **not** been added to the library yet." The fact that H5E-B teaches a single error stack is made clear in paragraph 4 (emphasis added):

> The Error interface provides error handling in the form of a stack. The FUNC_ENTER() macro clears the error stack whenever an interface function is entered. When an error is detected, an entry is pushed onto the stack. As the functions unwind, additional entries are pushed onto the stack. The API function will return some indication that an error occurred and the application can print the error stack.

There is no enabling description in the cited art for a multithreaded embodiment. To the contrary, H5E-B specifically states that it does not support multi-threading. Thus, while individual threads in single-threaded applications may have their own error stacks, H5E-B does not teach this for a multi-threaded program. In fact, the reference specifically states that it does not support multi-threading. The references do not described how a multi-threaded version of their teachings would work, or in any way provided an enabling disclosure for a multi-threaded version of their teachings. "In order to render a claimed apparatus or method obvious, the prior art must enable one skilled in the art to make and use the apparatus or method." *Motorola, Inc. v. Interdigital Tech. Corp.*, 43 USPQ2d 1481, 1489 (Fed. Cir. 1997) (quoting *Beckman Instruments, Inc. v. LKB Produkter AB*, 13 USPQ2d 1301, 1304 (Fed. Cir. 1989)); see also *Rockwell Int'l Corp. v. United States*, 47 USPQ2d 1027, 1032 (Fed. Cir. 1998). Since the cited art does not enable a multi-threaded version of its teachings (and in fact explicitly states that multi-threading is not supported), the rejection is improper.

Thus, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited prior art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 1 also apply to claims 9, 31, 36, 39, 47, 51, and 59.

Applicants also assert that the rejection of numerous ones of the dependent claims is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-69401/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicant

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: February 13, 2008